REMARKS

Claims 1 and 3-29 are pending in the Application. Previously, claims 1, 3, 6, 10, 14, 17-19, and 26 were amended, claim 2 was cancelled without prejudice or disclaimer, and claim 29 added, by a response filed April 24, 2008. Claim 1 is amended by the present response to correct a minor typographical error. Claims 1, 6, and 18 are independent claims, while claims 3-5 and 29, 7-17, and 19-28 depend either directly or indirectly from independent claims 1, 6 and 18, respectively.

The Applicant respectfully requests reconsideration of claims 1 and 3-29, in light of the following remarks.

Previous Rejection Under 35 U.S.C. §112

Applicant appreciates the Office Action's withdrawal of the previous rejection of claim 17 under 35 U.S.C. §112 for lack of antecedent basis. (*See* Office Action at p. 2)

Rejection of Claims Under 35 U.S.C. §103(a)

Claims 1 and 3-29 stand rejected under 35 U.S.C. §103(a) as being unpatentable over United States Patent No. 6,671,703, Thompson *et al*. (hereinafter "Thompson") in view of United States Patent No. 7,103,779 Kiehtreiber *et al*. (hereinafter "Kiehtreiber"). Applicant appreciates the Office Action's withdrawal of the previous anticipation and obviousness rejections. However, Applicant respectfully traverses the new grounds for rejection. As discussed below, Applicant respectfully submits that the presently claimed subject matter is allowable over the cited art.

Allowability of Claims 1 and Its Dependent Claims 3-5 and 29

Independent claim 1 recites a generator of difference information comprising, *inter alia*, an array storing operations for tree-based encoding of the first and second streams of information, wherein the generator outputs difference information including a differencing instruction set comprising a hierarchical tree map and a plurality of

operators represented by variable length codes based on frequency of occurrence of the associated operations. As known in the art, and discussed in the specification, a tree contains nodes linked between levels. The presently claimed subject matter relates to, *inter alia*, a generator that outputs difference information including a differencing instruction set comprising a hierarchical tree map. For example, in Fig. 1A, the "1" at the highest level node indicates that there is at least one "1" at the second level. The "1" at the second level node (second from the left in Fig. 1A) indicates that there is at least one "1" at the four nodes of the third level associated with that particular second level node. The "0" at the other three second level nodes indicates that there are no "1"s at any nodes on subsequent levels associated with those three particular second level nodes.

Applicant appreciates the Office Action's recognition that "Thompson does not expressly teach the use of a hierarchical tree encoding scheme." (*See* Office Action at p. 4). The Office Action, however, asserts that "Kiehtreiber teaches a method for incremental code signing comprising: a hierarchical tree encoding structure (Figures 3 and 4, also column 4, line 58 to column 5, line 7, where a program is broken into pages and a hash value is computed for each memory page, then another value is computed for the entire program based on the individual values)." The cited portion of Kiehtreiber (namely, 4:58-5:7) reads as follows:

> Next in steps 420 and 430, the system calculates a hash value for each memory page of the computer program 300 using a hash function 310. The hash function 310 may be any hash function such as the well-known SHA or MD5 has [sic] functions. As set forth in FIG. 3, the hash function 310 will create an associated hash value (390 to 389 [sic, 399]) for each memory page (380 to 389) of the computer program 300. The size of the output hash values in one embodiment are 20 bytes. However, many different sizes of hash values may be used.
>
> In step 440, the system of the present invention arranges the calculated hash values (390 to 389 [sic, 399]) into an array of hash values known as the hash array 373. The system then calculates an array hash value 360 for the

> entire hash array 373 using a hash function 370 in step 450. In one embodiment, hash function 370 is the same as hash function 310. However, a different hash function may be used.

Applicant respectfully submits that the cited disclosure of Kiehtreiber does not remedy the recognized deficiencies of Thompson. For example, the "hash array" of Kiehtreiber is not a tree, let alone a hierarchical tree. As discussed above, a tree has nodes linked between different levels. The mere disclosure of an "array" of hash functions does not teach, suggest, or otherwise render obvious the outputting of difference information including a differencing instruction set comprising an hierarchical tree map. The hash array, as described by Kiehtreiber, and illustrated in Fig. 3 of Kiehtreiber, does not include nodes linked between levels, let alone a hierarchical tree map.

The presently claimed subject matter of claim 1 for instance, outputs difference information including a differencing instruction set comprising a hierarchical tree map. Thus, the presently claimed subject matter includes a hierarchical tree map as part of its output. Kiehtreiber, on the other hand, discloses a hash array. The output of the asserted portion of Kiehtreiber is not a hierarchical tree map, let alone a differencing instruction set comprising a hierarchical tree map, but instead a program, a hash array, and a digital signature for a hash array: "Finally, at step 470, the hash array 373 and the digital signature for the hash array 350 are stored along the computer program 300. The hash array 373 and the digital signature for the hash array 350 are also distributed along with the computer program 300 such that any recipient of computer program 300 can verify its authenticity." (Kiehtreiber, 5:11-17). The cited portions of Kiehtreiber do not disclose the arrangement of that output comprising a hierarchical tree map.

For example, even if, *arguendo*, the hash values for each memory page were somehow construed as one level of a tree, and the hash value based on the individual values considered a different level of a tree, Kiehtreiber would still not disclose the presently claimed subject matter because the hash value based on the individual values is not output -- instead, a digital signature for that hash array is stored. Nor are the

different aspects of Figure 3 linked together as a hierarchical tree, as claimed by the presently claimed subject matter.

Moreover, the "pages" of the program and the hash values computed for each memory page of Kiehtreiber cannot disclose levels of a hierarchical tree. Each hash value is created from one memory "page" -- there is a one-to-one correspondence between these purported "levels" of a tree. Thus, even if, *arguendo*, they were somehow considered to be levels of a tree, they would still not disclose a hierarchical tree.

As a result of the foregoing, Applicant respectfully submits that Thompson and Kiehtreiber, either alone or in combination, do not teach, suggest, or otherwise render obvious the presently claimed subject matter, including "...wherein the generator... outputs the difference information between the first and second streams of information including a differencing instruction set comprising a hierarchical tree map ..."

Applicant further respectfully submits that, even if Kiehtreiber were combined with Thompson, the combination would still not teach, suggest, or otherwise disclose the claimed subject matter, as neither Kiehtreiber nor Thompson disclose, for example, "...wherein the generator... outputs the difference information between the first and second streams of information including a differencing instruction set comprising a hierarchical tree map ...," as discussed above. The combination still would not provide a differencing instruction set comprising a hierarchical tree map, as the asserted "hierarchical tree map" (which, from above, is not actually a hierarchical tree map) would not be part of any differencing instruction set. Instead, the asserted "hierarchical tree map" would just be used to authenticate whatever was sent in Thompson.

Applicant further respectfully submits that there are additional reasons why the cited combination does not render obvious claims dependent from claim 1. As one example, which will further underscore exemplary patentable distinctiveness of the presently claimed subject matter, Applicant notes that claim 5 recites the generator of claim 4 (which also depends from claim 1) "wherein the encoder employs variable

length encoding techniques for operators in a set of operations, the encoder employing tree-based variable sized blocks, and wherein the generator computes a cumulative address offset." The Office Action cites Kiehtreiber at 4:65-67 (which reads, "The size of the output hash values in one embodiment are 20 bytes. However, many different sizes of hash values may be used.") as disclosing that aspect of claim 5. Applicant respectfully traverses that rejection for at least two reasons.

First, the use of different sizes of hash values does not disclose "...tree-based variable sized blocks ..." as claimed in claim 5, because (in addition to not disclosing tree-based blocks as discusses above) it does not disclose the use of different sized hash values in the same embodiment. This is particularly so as Kiehtreiber teaches using one hash function for all of the memory pages, stating "the system calculates a hash value for each memory page of the computer program 300 using **a** hash function 310." (4:58-60; emphasis added). Kiehtreiber also states, "The size of the output hash values in one embodiment are 20 bytes." (4:65-66). Thus it appears that Kiehtreiber intends for each of the hash values in any one embodiment to be the same, and therefore does not teach tree-based variable sized blocks. Applicant respectfully submits that using a same or different hash function on the hash array would not remedy that deficiency in the teaching of Kiehtreiber.

Second, even if the hash function of Kiehtreiber did somehow disclose tree-based variable sized blocks, it still would not disclose the presently claimed subject matter. This is because claim 5 recites "wherein the encoder employs variable length encoding techniques for operators in a set of operations, the encoder employing tree-based variable sized blocks, and wherein the generator computes a cumulative address offset." The memory pages in the cited portion of Kiehtreiber are not operators, but instead portions of a program to be verified.

As a result of the at least the foregoing, Applicant respectfully submits that Thompson and Kiehtreiber, either alone or in combination, do not teach suggest, or otherwise render obvious claim 1 or any of its dependent claims, and that those claims

are allowable. Applicant further respectfully submits that the Office Action fails to present a *prima facie* case of obviousness. As a result, Applicant respectfully requests withdrawal of the rejections of those claims.

Allowability of Claim 6 and Its Dependent Claims 7-17

Independent claim 6 recites an electronic device network adapted to dispense streaming updates to at least one of a plurality of electronic devices comprising, *inter alia*, a generator generating streaming updates, the generator processing at least one of a plurality of blocks of content, the at least one of a plurality of blocks of content comprising a stream of bytes, the generator processing the at least one of a plurality of blocks of content until reaching an end of the stream of bytes, the generator comprising an encoder employing a tree-based hierarchy for encoding a block of operations." From the above discussion, the cited combination does not teach, suggest, or otherwise render obvious such an encoder employing a tree-based hierarchy for encoding a block of operations. As such, Applicant respectfully submits that Thompson and Kiehtreiber, either alone or in combination, do not teach, suggest, or otherwise render obvious claim 6 or any of its dependent claims, and that those claims are allowable. Applicant further respectfully submits that the Office Action fails to present a *prima facie* case of obviousness for those claims. As a result, Applicant respectfully requests withdrawal of the rejections of those claims.

Allowability of Claim 18 and Its Dependent Claims 19-28

Independent claim 18 recites a method of generating streaming updates by converting a first stream of information into a second stream of information for updating an electronic device, the method comprising, *inter alia*, "creating a hierarchical tree-based transform output from operators determined in the transform, wherein the hierarchical tree-based transform output comprises at least three levels for encoding a

block of N operations, the hierarchy comprising a top level wherein each node of the top level encodes N bytes, a second level wherein each node of the second level encodes N/4 bytes, and a third level wherein each node of the third level encodes N/16 bytes." Applicant respectfully submits that, for the reasons discussed previously, the cited combination does not teach, suggest, or otherwise render obvious a method comprising, *inter alia*, "creating a hierarchical tree-based transform output from operators determined in the transform…" As a result, Applicant submits that claim 18 and its dependent claims are allowable.

Claim 19 further recites that the tree-based transform output comprises "at least three levels for encoding a block of N operations, the hierarchy comprising a top level wherein each node of the top level encodes N bytes, a second level wherein each node of the second level encodes N/4 bytes, and a third level wherein each node of the third level encodes N/16 bytes." The Office Action asserts Kiehtreiber discloses "a tree based hierarchy employed by the encoder comprising at least three levels for encoding a block of N operation, the hierarchy comprising a top level wherein each node of the top level encodes N bytes, a second level wherein each node of the second level encodes N/4 bytes, and a third level wherein each node of the third level encodes N/16 bytes" because, *inter alia*, "The encoding [of Kiehtreiber] therefore uses a three level encoding scheme, where the program/page, then page hash value, then array hash value, where each is smaller than the previous." (Office Action at p. 11). Applicant respectfully submits that to say the occupants of one asserted "level" are larger or smaller than those at another misses the point – it is how many bytes are being encoded that is claimed. This difference between Kiehtreiber and the claimed subject matter further underscores that Kiehtreiber does not disclose an hierarchical tree. For example, each page of the asserted "program/page" level directly corresponds to one hash value in the asserted "page hash value." To the extent they are argued to encode anything, they are "encoding" the same thing: one page. Thus, they cannot be encoding different numbers of bytes, let alone, for example one level encoding N/4 bytes per node and the next level encoding N/16 bytes per node. This provides an

additional reason for the allowability of claim 18. (*See* also claims 10 and 29 which have a related limitation.)

As one example of the additional allowability of a claim dependent from claim 18, Applicant notes that claim 26 recites, *inter alia*, "encoding a node and sub-nodes in a way indicating an impossible difference is employed as an escape sequence during encoding." The current Office Action asserts that Thompson teaches this aspect at 3:43-46, which states, "Generally, the file difference synchronization method will go along comparing bytes in both files, as long as they match, the count is increased, which will be the amount for a skip record." Applicant respectfully submits that, even if, *arguendo*, such a "skip record" or "skip count" may provide an escape sequence (as the Office Action asserts) such a "skip record" or "skip count" does not teach, suggest, or otherwise render obvious "encoding a node and sub-nodes in a way indicating an impossible difference is employed as an escape sequence during encoding." As such, Applicant respectfully submits this provides an additional reason for the allowability of claim 26.

As a result of at least the foregoing, Applicant respectfully submits that Thompson and Kiehtreiber, either alone or in combination, do not teach, suggest, or otherwise render obvious claim 18 or any of its dependent claims, and that those claims are allowable. Applicant further respectfully submits that the Office Action fails to present a *prima facie* case of obviousness for those claims. As a result, Applicant respectfully requests withdrawal of the rejections of those claims.

Conclusion

In general, the Office Action makes various statements regarding claims 1 and 3-29 and the cited references that are now moot in light of the above. Thus, Applicants will not address such statements at the present time. However, Applicants expressly reserve the right to challenge such statements in the future should the need arise (e.g., if such statements should become relevant by appearing in a rejection of any current or future claim).

Applicants believe that all of claims 1 and 3-29 are in condition for allowance. Should the Examiner disagree or have any questions regarding this submission, the Applicants invite the Examiner to contact the undersigned at (312) 775-8000 for an interview.

A Notice of Allowability is courteously solicited.

Respectfully submitted,

Date: August 4, 2008

/Kevin E. Borg/
Kevin E. Borg
Reg. No. 51,486

Hewlett-Packard Company
Intellectual Property Administration
Legal Department, M/S 35
P.O. Box 272400
Fort Collins, CO 80527-2400